

University of Groningen

Assessing Schizophrenia with an Interoperable Architecture

Emerencia, Ando; van der Krieke, Lian; Petkov, Nicolai; Aiello, Marco

Published in:

Proceedings of the first International Workshop on Managing Interoperability and Complexity in Health Systems, MIXHS'11

DOI:

[10.1145/2064747.2064766](https://doi.org/10.1145/2064747.2064766)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2011

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Emerencia, A., van der Krieke, L., Petkov, N., & Aiello, M. (2011). Assessing Schizophrenia with an Interoperable Architecture. In M-M. Bouamrane, & C. Tao (Eds.), *Proceedings of the first International Workshop on Managing Interoperability and Complexity in Health Systems, MIXHS'11* (pp. 79-82). ACM Press. <https://doi.org/10.1145/2064747.2064766>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Assessing Schizophrenia with an Interoperable Architecture

Ando Emerencia
University of Groningen
a.c.emerencia@rug.nl

Nicolai Petkov
University of Groningen
n.petkov@rug.nl

Lian van der Krieke
University Medical Center
Groningen
j.a.j.van.der.krieke@umcg.nl

Marco Aiello
University of Groningen
m.aiello@rug.nl

ABSTRACT

With the introduction of electronic personal health records and e-health applications spreading, interoperability concerns are of increasing importance to hospitals and care facilities. Interoperability between distributed and complex systems requires, among other things, compatible data formats. The recommended approach is to store data using international terminology standards. For data that is not stored in this way, a conversion process must happen. This can be tedious manual work when multiple input and output formats are to be supported. We present WEGWEIS, a web application for schizophrenia patients that converts questionnaire answers into advice. The system's advice delivery is based on data extracted from the electronic medical records of 1379 patients. In WEGWEIS, we handle the conversion by decoupling input formats from output formats, using an ontology as intermediate layer. We present the algorithm and provide details on its implementation.

Categories and Subject Descriptors

J.3 [Computing Applications]: Life and Medical Sciences—*Medical information systems*; H.4.m [Information Systems Applications]: Miscellaneous

General Terms

Algorithms, Design

Keywords

Case-based reasoning, ontology, routine outcome monitoring

1. INTRODUCTION

While the benefits of applying data mining and machine learning techniques to medical data have been mostly in

favor of the clinicians, currently there is a surge of patient-centered applications. Through personal websites, patients are given more control over their treatment. A key functionality of these websites is to inform the patients about their disease and treatment options [7]. Consider the specific case of schizophrenia. For people with such a disease or other severe mental illnesses, little has been achieved thus far [5]. Schizophrenia is a mental disorder that affects approximately 1% of the population and is identified by psychoses: episodes where grip on reality is lost. The symptoms are caused by impaired processing of information in the brain. Due to the cognitive problems affecting these patients (e.g. concentration problems), they are easily overwhelmed by too much information [4] and thus for any application to be effective it must be able to select and present the patient only what is actually relevant for him/her. Typical case-based reasoning approaches do not work in a context where the outcomes (i.e. the information shown to a patient) are assumed to be fully dynamic in that they can be added or removed on the fly. The need is for an approach that promotes interoperability so that the knowledge gained can easily be reused in different contexts.

In this paper, we describe the data conversion algorithm of WEGWEIS, a web application that can filter chunks of relevant information ('advice') for individual schizophrenia patients, based on information from their electronic medical records. WEGWEIS breaks with conventional case-based reasoning approaches in that it decouples symptoms from outcomes (the information chunks), allowing the latter to be dynamic (i.e. adding or removing advice/outcomes on the fly). The system maintains connections between the symptom and outcome layers through an ontology that allows seamless extensibility by classifying problems as special instances of more generic ones, and through the transitive property advice can be inferred for any problem or category of problems. WEGWEIS distinguishes important problems from common ones by analyzing questionnaire answers from 1379 patients. We plan to assess the efficacy of WEGWEIS in a randomized clinical trial.

The rest of the paper is organized as follows. Section 2 discusses recent developments that form the motivation for our approach. In Section 3, we illustrate the WEGWEIS system design. Section 4 explains the details of the ontology mapping algorithm, with an example scenario given in Section 5. Section 6 concludes the paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MIXHS'11, October 28, 2011, Glasgow, Scotland, UK.

Copyright 2011 ACM 978-1-4503-0954-7/11/10 ...\$10.00.

2. BACKGROUND

Nowadays there is a trend from electronic health/medical records to personal health records, which increases complexity. Personal health records are managed by the patients themselves instead of by practitioners, and are located remotely rather than at a specific hospital.

To improve care, these different services need to be able to cooperate. This gives rise to new challenges. For example there are many different (often locally customized) implementations of electronic health records. Standards have been developed as a requirement for the interoperability of health care applications, including messaging formats (HL7 v2.x, HL7 v3.x, ISO13606), patient summaries (HL7 CDA, CCR, CCD) and terminology (GALEN, UMLS, LOINC, SNOMED-CT, DICOM – for images).

Medical ontologies give background knowledge (interpretations, relations) to data expressed using these standards. Dietterich et al. [2] stress the need for dealing with background knowledge using ontologies. There are numerous examples of ontology-driven health care systems that are used in practice. Blobel and Oemig [1] describe architectures of such systems and note that in the case of complex systems or system integration, ontologies can be harmonized using a higher level ontology. In WEGWEIS we show how these concepts can apply not only to standardized data formats, but also to seemingly unstructured medical data such as questionnaire answers and advice.

3. WEGWEIS DESIGN

WEGWEIS is a web application to support schizophrenia patients by offering relevant information based on questionnaire answers. In essence it takes information intended for clinicians, it reasons on it and makes it understandable and accessible to patients.

Routine outcome monitoring is a procedure used in psychiatric health care where patients periodically fill out a number of questionnaires using an online questionnaire manager. Figure 1 shows how WEGWEIS interfaces with such a questionnaire manager to obtain data stored in electronic health records. Through the use of a problem ontology the questionnaire answers are interpreted and the information obtained is converted into personalized advice or other output formats, e.g. for use in personal health records.

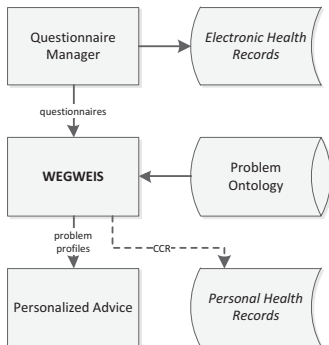


Figure 1: WEGWEIS system design.

To populate the ontology, we analyzed the following questionnaires: the Health of Nations Outcome Scale (HoNOS, HoNOSCA), the Manchester Short Assessment (MANSA)

and the OQ-45 (Outcome Questionnaire). We constructed this ontology in cooperation with experts based on the questionnaires that are in use in the University Medical Center Groningen. It currently contains 117 nodes. We create ontology mappings for these questionnaires such that each question determines the applicability of a problem node, as shown in Figure 2.

Most questionnaire answers are quantitative measures: they express the severity of how much the patient is affected by the problem. A measure of *problem importance* is established by applying a normalization procedure to each questionnaire answer. After which, we define a global minimum threshold for triggering the applicability of a problem (e.g. 0.5).

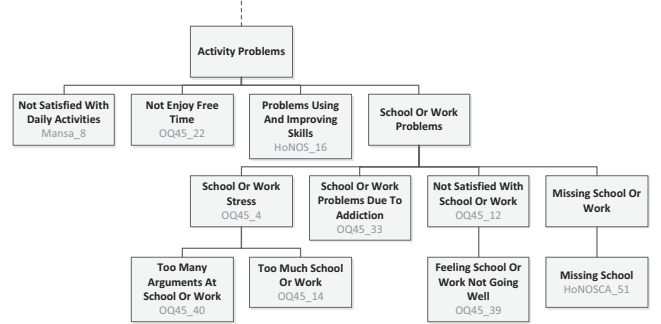


Figure 2: Part of the ontology. Some nodes are linked to questionnaire fields.

The approach is structured around an ontology, where each node represents a problem. The specificity and applicability of problem nodes in the ontology follow from its structure. The ontology is hierarchical in that the child nodes are special cases of their parent node. More formally, ground mereology [3] states that this ontology is a relation that captures a partial order that is reflexive, antisymmetric and transitive, with respect to the *instance of* relationship.

For example see Figure 2, which shows part of the ontology. The transitivity property says that if **MissingSchoolOrWork** is a **SchoolOrWorkProblem**, which is an **ActivityProblem**, then **MissingSchoolOrWork** is also an **ActivityProblem**, which follows our intuition. Such rules allow us to derive properties of *specificity* and *applicability*.

As for the *applicability* property, we know that if a patient suffers from a certain problem, then all ancestor nodes of this problem apply to the patient as well, since the child nodes are their special cases. Hence given a set of problems (nodes) for a patient, we can derive the full set of applicable problems by including all ancestors of all nodes in that set.

The *specificity* property is obvious for two nodes where one is an ancestor of the other, this expresses relative specificity. In the algorithm explained in the next section we extend on this concept by also defining absolute specificity, which looks at the minimum distance to a leaf node. This is a weighted distance, where each edge going to a node that is included in the output mapping has a weight of 1, and edges to other nodes have a weight of 0.

4. ONTOLOGY MAPPING

The ontology mapping conversion algorithm converts questionnaire answers for a patient into ordered lists of problems (severe and specific problems first), through the use of

an ontology. Translating data obtained from the user into a workable expert advice. The Advice \leftrightarrow Problem mapping and the advice units were created with the help of psychiatrists. The psychiatrists were asked what advice they would give for a certain problem, and to which other problems the same advice might apply. There are no exclusion criteria for advice, as leaving out key advice is considered more harmful than giving too much advice. While the problem ontology itself is not consumer-oriented, the algorithm can support consumer health vocabularies as in- or output formats. We explain the algorithm in the context of giving advice to the patient, though we remark its generality as it can easily be used for other purposes. By replacing the Advice \leftrightarrow Problem mapping with a SNOMED-CT ID \leftrightarrow Problem mapping for example, and using the results to generate a CCR.

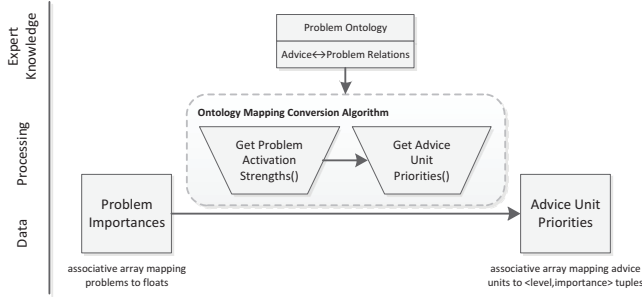


Figure 3: Information flow in WEGWEIS.

The algorithm starts with a set of problems and their importance; it outputs a set of advice units and their priorities (the priorities can be used to imply an ordering). The algorithm extends the given outcome ontology mapping (Advice \leftrightarrow Problem relations in Figure 3) by using the problem ontology in order to infer advice when a specific instance of an applicable problem is found. The algorithm consists of two steps, namely (i) calculating the activation strengths and (ii) using those to calculate the advice unit priorities. We describe these steps next.

An activation strength is a measure that combines specificity with importance. The activation strength for a problem p is calculated as the maximum augmented activation strength of p and its descendants, where the augmentation for a descendant q consists of decreasing the specificity for every advice unit that applies to q but not to p .

The algorithm shown in Figure 4 starts by initializing P to be the set of all problems in the ontology and T to problem importances with level 0. Note that T and A hold intermediate results; B is eventually returned. The outer loop traverses over all nodes in P , every iteration taking the set of leaf nodes and removing them. In the inner loop $T[p]$ is set to the maximum of itself and its descendant nodes, and if its value is not null then it is copied to $B[p]$. After all leaf nodes in the current iteration have been considered, T and A are updated to account for advice given in this iteration.

The algorithm GETLEAFNODES shown in Figure 5 returns the subset of relative leaf nodes within the given set P . These are the nodes that have no descendant nodes in P . It loops over all the problems in P and returns those problems whose sets of descendants according to the ontology have no elements in common with P .

```

Algorithm      GETPROBLEMACTIVATIONSTRENGTHS( $V$ )

Input:    associative array  $V$  mapping problems to problem importances
              (floats).
Data:    ontology functions all_problems and descendants.
Output:   associative array mapping problems to  $\langle level, importance \rangle$  tuples,
              for all triggered problems.

 $P \leftarrow \text{all\_problems}()$ 
 $B \leftarrow$  empty associative array
 $T \leftarrow$  empty associative array
 $A \leftarrow$  empty associative array
for each problem  $p \in V.\text{keys}$ 
  do  $T[p] \leftarrow \langle 0, V[p] \rangle$ 
while  $P$  is not empty
   $N \leftarrow \text{GETLEAFNODES}(P)$ 
  for each problem  $p \in N$ 
    for each problem  $q \in \text{descendants}(p)$ 
      do if  $T[q]$ 
        then  $T[p] \leftarrow \max(T[p], T[q])$ 
    do if  $T[p]$ 
      then  $B[p] \leftarrow T[p]$ 
      remove  $p$  from  $P$ 
   $T, A \leftarrow \text{UPDATEPROBLEMLEVELS}(N, T, A)$ 
return ( $B$ )

```

Figure 4: The GetProblemActivationStrenghts algorithm.

```

Algorithm      GETLEAFNODES( $P$ )

Input:    set of problems  $P$ .
Data:    ontology function descendants.
Output:   the subset of problems that are relative leaf nodes.

 $L \leftarrow$  empty set
for each problem  $p \in P$ 
  do if (descendants( $p$ )  $\cap P$ ) is empty
    then add  $p$  to  $L$ 
return ( $L$ )

```

Figure 5: The GetLeafNodes algorithm.

The algorithm UPDATEPROBLEMLEVELS shown in Figure 6 updates the problem levels after each iteration, for every advice unit given by N it decreases the level of all problems that it applies to (and that of their descendants). The result is that a problem that is covered by an advice unit will trigger advice in later iterations (i.e. more generic advice) with lower activation strength. The algorithm first sets U to be the set of all advice units that are triggered by nodes in N . Then for each advice unit, it tries to decrease the level of all problems that it applies to (i.e. all problems that directly trigger it and all their descendants). Some bookkeeping is done in A to ensure that one advice unit does not decrease a level multiple times (e.g. over different iterations).

The algorithm GETADVICEUNITPRIORITIES shown in Figure 7 outputs a set of applicable advice units along with their priorities. The priority of an advice unit is the maximum activation strength of the problem nodes that directly trigger the advice unit.

The approach as a whole favors giving specific advice since GETPROBLEMACTIVATIONSTRENGTHS handles leaf nodes first, and favors giving diverse advice since in UPDATEPROBLEMLEVELS the levels of all problems that an advice unit applies to are decreased.

Algorithm **UPDATEPROBLEMLEVELS**(N, T, A)

Input: set of problems N , associative array T mapping problems to $\langle \text{level}, \text{importance} \rangle$ tuples, associative array A mapping problems to lists of advice units.

Data: ontology function `descendants`, function `advice_triggered_by`, function `problems_triggering`.

Output: updated T and A , where the mappings have been updated to reflect advice given by N .

```

 $U \leftarrow$  empty set
for each problem  $p \in N$ 
  do if  $T[p]$ 
    then {
      for each advice unit  $a \in \text{advice\_triggered\_by}(p)$ 
        do add  $a$  to  $U$ 
    }
for each advice unit  $u \in U$ 
  {
    for each problem  $p \in \text{problems\_triggering}(u)$ 
      do {
        for each problem  $q \in (\{p\} \cup \text{descendants}(p))$ 
          do if  $T[q]$  and not  $u \in A[q]$ 
            then {
               $\langle l, s \rangle \leftarrow T[q]$ 
               $T[q] \leftarrow \langle l - 1, s \rangle$ 
               $A[q] \leftarrow A[q] \cup \{u\}$ 
            }
          }
      }
  }
return ( $T, A$ )

```

Figure 6: The UpdateProblemLevels algorithm.

Algorithm **GETADVICEUNITPRIORITIES**(B)

Input: associative array B mapping problems to $\langle \text{level}, \text{importance} \rangle$ tuples (i.e. `GETPROBLEMACTIVATIONSTRENGTHS()`).

Data: function `advice_triggered_by`.

Output: associative array mapping advice units to $\langle \text{level}, \text{importance} \rangle$ tuples.

```

 $R \leftarrow$  empty associative array
for each problem  $p \in B.\text{keys}$ 
  do {
    for each advice unit  $a \in \text{advice\_triggered\_by}(p)$ 
      do  $R[a] \leftarrow \max(R[a], B[p])$ 
  }
return ( $R$ )

```

Figure 7: The GetAdviceUnitPriorities algorithm.

5. EXAMPLE SCENARIO

Figure 8 shows an example scenario where we have three nodes, one of them has an advice attached to it (“Talk to Case Manager”), and two of them were answered above the threshold (0.67 and 0.75). Initially in `GETPROBLEMACTIVATIONSTRENGTHS` we set $T = \{\alpha \Rightarrow \langle 0, 0.67 \rangle, \gamma \Rightarrow \langle 0, 0.75 \rangle\}$.

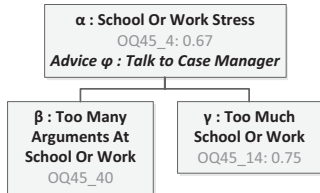


Figure 8: An example scenario with three nodes.

The first iteration finds $N = \{\beta, \gamma\}$ as leaf nodes. Since neither of these nodes have descendants, T remains unchanged in the first inner loop. B becomes $\{\gamma \Rightarrow \langle 0, 0.75 \rangle\}$. Nothing happens in the call to `UPDATEPROBLEMLEVELS`, since none of the nodes in N directly trigger an advice.

In the second iteration we find $N = \{\alpha\}$, and T becomes $\{\alpha \Rightarrow \langle 0, 0.75 \rangle, \gamma \Rightarrow \langle 0, 0.75 \rangle\}$ since γ is a **descendant** of α . These are also the values returned by B . After the second iteration, `UPDATEPROBLEMLEVELS` sets A to

$\{\alpha \Rightarrow \varphi, \gamma \Rightarrow \varphi\}$, and T to $\{\alpha \Rightarrow \langle -1, 0.75 \rangle, \gamma \Rightarrow \langle -1, 0.75 \rangle\}$, signifying that an advice φ has been given that applies to these problems.

6. CONCLUDING REMARKS

We presented an algorithm that is used for converting questionnaire answers to advice. The algorithm is the core of WEGWEIS, a web application for patients. In [6], we have explained the utility of our system from the medical point of view. Beyond the specific application of schizophrenia management, our approach can be abstracted to work in any task relating to converting medical data, since all domain knowledge is contained in the ontology. Ontology based data conversion requires less effort when multiple formats are involved since formats only need to be tied to the ontology and not to each other. Inferring over the structure of the ontology furthermore allows extensions and additions to be integrated seamlessly.

Given our proposal for ontology-based data conversion in e-health applications for schizophrenia patients, we envision that any medical data can automatically be made accessible and understandable for any patient. To achieve this vision the technical challenges we foresee are (i) to standardize not only diseases and symptoms in ontologies for clinicians, but also advice and explanations for patients; (ii) to achieve interoperability for complex health systems through ontology-based data conversion; and (iii) to test these approaches on large data sets and with feedback from patients.

7. ACKNOWLEDGMENTS

The research is supported by the Dutch National Research Council for Innovation in Health Care (ZonMW) under contract no. 300020011. We thank S. Sytema and D. Wiersma for fruitful discussion.

8. REFERENCES

- [1] B. Blobel and F. Oemig. Ontology-driven health information systems architectures. *Studies in health technology and informatics*, 150:195, 2009.
- [2] T. Dietterich, P. Domingos, L. Getoor, S. Muggleton, and P. Tadepalli. Structured machine learning: the next ten years. *Machine Learning*, 73(1):3–23, 2008.
- [3] C. Keet and A. Artale. Representing and reasoning over a taxonomy of part-whole relations. *Applied Ontology*, 3(1):91–110, 2008.
- [4] A. Rotondi et al. Designing websites for persons with cognitive deficits: Design and usability of a psychoeducational intervention for persons with severe mental illness. *Psychological Services*, 4(3):202, 2007.
- [5] M. Välimäki et al. Design and development process of patient-centered computer-based support system for patients with schizophrenia spectrum psychosis. *Inf. for Health and Social Care*, 33(2):113–123, 2008.
- [6] L. van der Krieke, A. Emerencia, and S. Sytema. An online portal on outcomes for dutch service users. *Psychiatric Services*, American Psychiatric Association. To appear.
- [7] H. Wald, C. Dube, and D. Anthony. Untangling the web—the impact of internet use on health care and the physician-patient relationship. *Patient education and counseling*, 68(3):218–224, 2007.